LAB 15

Objective: Install Helm and launch an application packaged as a Chart.

Download Helm from the GitHub release page. Verify that the **helm** binary is in your PATH:

```
$ helm version
```

From your local machine, you can setup Helm on minikube. **helm init** will launch a **tiller-deploy** deployment.

```
$ helm init
$ kubectl get deployment --all-namespaces
NAMESPACE     NAME             DESIRED    CURRENT    UP-TO-DATE    AVAILABLE
AGE
kube-system   tiller-deploy    1          1          1             1
2m
```

You are ready to search for Charts and install them in your cluster. Let's search for *redis* and install the corresponding chart. We will not use persistence for this exercise.

```
$ helm search redis
NAME                          VERSION DESCRIPTION
stable/redis                  0.4.2   Chart for Redis
$ helm install stable/redis --set persistence.enabled=false
$ helm list
NAME            REVISION    UPDATED                         STATUS      CHART
```

```
misty-lionfish  1           Tue Dec 20 19:01:18 2016    DEPLOYED
redis-0.4.2
```

With a deployed Chart, you should see a *redis* pod being created.

```
$ kubectl get pods
NAME                                    READY     STATUS    RESTARTS   AGE
misty-lionfish-redis-3119530502-3q0xl   1/1       Running   0          10s
```

Helm uses ConfigMaps to store each release information. Check your ConfigMaps. You can then delete your release.

```
$ kubectl get configmap --all-namespaces
NAMESPACE     NAME                  DATA      AGE
kube-system   misty-lionfish.v1     1         5m
$ helm delete misty-lionfish
```

This is just the quickest exercise with Helm. You can now dive into writing your own Chart, setting up your own repository and using Helm to do rolling updates of entire applications.